

PATENT

CERTIFICATE OF MAILING VIA "EXPRESS MAIL"

PURSUANT TO 37 C.F.R. 1.10, I HEREBY CERTIFY THAT THIS PAPER OR FEE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" AND IS ADDRESSED TO:

THE HONORABLE COMMISSIONER OF PATENTS  
BOX PATENT APPLICATION  
WASHINGTON, D.C. 20231

ON MARCH 30, 2001.



Ronald L. Chichester Reg. No. 36,765

Date: 03.30.01

Express Mail:

Express Mail Label: EL477922109US

APPLICATION FOR LETTERS PATENT

FOR

DEBUGGING EMBEDDED SYSTEMS

Inventors: Charles Simmers and Joseph W. Triece

ASSIGNEE: MICROCHIP TECHNOLOGY INCORPORATED

## DEBUGGING EMBEDDED SYSTEMS

### Field of the Invention

The present invention relates generally to background debuggers, and more particularly to on-chip debuggers, and even more particularly to on-chip debuggers for microcontrollers.

### Background of the Invention Technology

In recent years, microprocessors have become almost commonplace in electronic devices. Indeed, even household appliances, such as washing machines, refrigerators and water heaters, may include microprocessors to control some aspect of their operation. A microprocessor used in such an application is frequently referred to as a "microcontroller." An application that incorporates a microprocessor is sometimes called an "embedded system," because the control for the system is embedded in the system rather than being external to the system.

One of the challenges of testing embedded systems is that the microcontroller and the system it is controlling are frequently so intertwined as to make testing the microcontroller apart from the system, or vice versa, very difficult. In the past, engineers have addressed this problem by using "in-circuit emulators," or ICEs, which are sophisticated systems that emulate the operation of the microprocessor through a cable and connector that connect in the place usually occupied by the microprocessor.

ICE systems typically allow a user to set "breakpoints" in the microprocessor code. Generally, the breakpoints are associated with a condition of the microprocessor, such as a program address being accessed, a data address being accessed or some other event. The ICE executes software that emulates execution of the microprocessor code 5 and monitors the condition of the emulated microprocessor to detect breakpoints. When a breakpoint is reached, the ICE stops emulating the microprocessor operation and allows the condition of the emulated microprocessor to be examined. Assuming that the emulator software is operating correctly and that it correctly emulates the microprocessor operation, an ICE system can be used to debug the hardware and software of embedded 10 microprocessor systems.

What is needed is a debugger that will work without removing the embedded microprocessor and thus will not rely on the accuracy of an emulation.

#### SUMMARY OF THE INVENTION

The invention overcomes the above-identified problems as well as other 15 shortcomings and deficiencies of existing technologies by providing a debugging system that works without removing the embedded microprocessor. Registers provided with the system store breakpoint conditions. Logic coupled to the registers and the system busses determines when the monitored conditions occur and interrupts the system. The registers and logic are implemented in hardware which allows an embedded system to be 20 debugged with the microprocessor or microcontroller installed in the system and running its own software or firmware. In one embodiment, the debug logic is implemented on the

same chip as the microcontroller or microprocessor. In another embodiment, the debug logic is incorporated in a separate module that can be coupled to the microprocessor or microcontroller for debugging purposes.

In accordance with an exemplary embodiment of the present invention, an  
5 embedded system is provided with the capability to be debugged. The embedded system includes a central processing unit (CPU) that is coupled to a bus having certain contents. A register, also with contents, is available for loading by the CPU. Finally, a debug logic circuit is also included. The debug logic circuit is coupled to both the bus and the CPU.  
The debug circuit itself is composed of a breakpoint detect circuit that is coupled to the  
10 bus and to the register. This circuitry enables a breakpoint signal that is produced by the breakpoint detect circuit when the contents of the register equal the contents of the bus.

In another embodiment of the present invention, a method is disclosed for debugging an embedded system having a microcontroller with a CPU. First, a debug logic circuit that resides on the same chip as the microcontroller is programmed to detect  
15 a predetermined condition in the microcontroller. Next, application software is run on the microcontroller. When a predetermined condition is detected, the CPU is interrupted which provides the ability to view the condition of the microcontroller. Programming the debug logic circuit can include the storing of a breakpoint address in a breakpoint address register. Afterward, a program memory address bus is selected for comparison to the  
20 contents of the breakpoint address register, upon which time a breakpoint counter is set to zero. The steps of interrupting and detecting are accomplished by comparing the contents

of the program memory address bus to the contents of the breakpoint register and, if they are equal, then the CPU is interrupted.

Yet another embodiment of the present invention is composed of a bus interface for interfacing to a microcontroller bus. In addition, a communications interface is 5 included for receiving debug instructions. A register is also provided which holds the contents that can be loaded through the communications interface. A breakpoint detect circuit is coupled to the bus interface and the register. In operation, a breakpoint signal is produced by the breakpoint detect circuit when the contents of the register equal the contents of the bus.

10 A technical advantage of the present invention is that the debugger can be executed with the microcontroller in place in the target system and executing target system code. This eliminates reliance on ICE interpreting target system code.

Features and advantages of the invention will be apparent from the following description of the embodiments, given for the purpose of disclosure and taken in 15 conjunction with the accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present disclosure and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, wherein:

20 Figure 1 is a block diagram of a target system in a debug configuration.

Figure 2 is a block diagram of a target system.

Figure 3 is a block diagram of a prior art microcontroller.

Figure 4 is a block diagram of a microcontroller including on-chip debug logic.

Figure 5 is a block diagram of a microcontroller and a separate debug module in a

5 debug configuration.

Figure 6 is a map of the debug registers.

Figure 7 is a block diagram of the debug logic.

Figure 8 is a flow chart for setting up a break on program memory address match.

Figure 9 is a flow chart for monitoring for a program memory address match

10 breakpoint.

Figure 10 is a flow chart for setting up a break on number of program memory address matches.

Figure 11 is a flow chart for monitoring for a number of program memory address matches breakpoint.

15 Figure 12 is a flow chart for setting up a break on SFR address match.

Figure 13 is a flow chart for monitoring for a SFR address match breakpoint.

Figure 14 is a flow chart for setting up a break on SFR address and data match.

Figure 15 is a flow chart for monitoring for a SFR address and data breakpoint.

Figure 16 is a flow chart for setting up a break on number of SFR address and data matches.

5       Figure 17 is a flow chart for monitoring for a number of SFR address and data matches breakpoint.

While the present invention is susceptible to various modifications and alternative forms, specific exemplary embodiments thereof have been shown by way of example in the drawings and are herein described in detail. It should be understood, however, that  
10      the description herein of specific embodiments is not intended to limit the invention to the particular forms disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

#### DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

15       The present invention is directed to debugging embedded systems.

Referring now to the drawings, the details of an exemplary embodiment of the present invention is schematically illustrated. Like elements in the drawings will be represented by like numbers, and similar elements will be represented by like numbers with a different lower case letter suffix.

In an exemplary embodiment, illustrated in Fig. 1, a target system under development 100 is configured to be debugged under the control of a personal computer ("PC") 105 running a development tool. An example of such a development tool is the MPLAB system, available commercially from Microchip Technology Incorporated.

5       A protocol translator gateway 110 contains a microcontroller which translates the commands from the PC 105 into a format acceptable to the microcontroller. In the case of microcontrollers manufactured by Microchip Technology Incorporated, the interface is a serial interface. The serial interface allows software running on the PC 105 to control the debugging of the target system 100.

10       In one exemplary embodiment, the target system under development, shown in more detail in Fig. 2, includes a microcontroller 200 with a serial interface to the protocol translator gateway 110. The microcontroller 200 also includes outputs 205 for controlling actuators in a controlled system 210 and inputs 215 for monitoring sensors installed in the controlled system 210. The controlled system 210 can be any kind of  
15       system susceptible to control in this manner. For example, the controlled system 210 can be a household appliance, such as a refrigerator or a water heater.

An existing microcontroller 200, illustrated in Fig. 3, includes a CPU 300, which communicates with the protocol translator gateway 110 through a serial I/O interface 305. The CPU controls actuators and reads sensors on the controlled system 210 through  
20       a sensor/actuator I/O 310. The microcontroller 200 includes a program memory 315 and a data memory 320, connected by separate busses to the CPU 300.

In one exemplary embodiment, shown in Fig. 4, the microcontroller 200 is modified to include a debug logic circuit 400. The debug logic circuit 400 interfaces with the program memory busses and the data memory busses and produces a break signal 405. The break signal 405 is coupled to a CPU input so that when the break signal 405 is activated, the CPU will be interrupted and vectored in a conventional manner to an debug interrupt vector where code is stored for executing appropriate interrupt software or firmware. In one exemplary embodiment, the data memory 320 includes a set of debug registers 410 which facilitate debugging, as described below.

In an alternative embodiment, illustrated in Fig. 5, the debug logic circuit 400 and registers 410 are located in a module 500 separate from the microcontroller. The module 500 has a serial interface which allows it to be controlled by the PC 105 through the protocol translator gateway 110. The module 500 also has an interface that allows it to connect to the microcontroller's program memory busses and data memory busses and to provide the break signal 405, as described above.

The embodiment illustrated in Fig. 4 has the advantage that it is always available to be debugged. The disadvantage is that some space on the microcontroller chip must be devoted to the debug logic circuit. This disadvantage is mediated by the simplicity of the debug logic circuit, as shown below.

The advantage of the embodiment illustrated in Fig. 5 is that it can be applied to any microprocessor or microcontroller system in which the busses are available as shown in Fig. 5 and in which the CPU 300 has an interrupt input that can be used for debug

purposes. The disadvantage of this embodiment is that a separate debug module is necessary.

The debug registers used in one exemplary embodiment, illustrated in Fig. 6, include a DEBUG register 600, an ICKBUG register 605, a BIGBUG register 610, a 5 GLOBUG register 615, a CNTBUG register 620 and a SLGBUG register 625. A configuration register (not shown) includes a BKBUG bit which, if set to zero, enables the background debugger hardware.

The DEBUG register includes a read-only INBUG bit which is set to "1" when the device is executing background debugger code. A FREEZ bit, when set to "1," will 10 cause peripherals to freeze when INBUG is set to "1." When a SSTEP bit is set to "1," the debug program will execute one instruction word of user code upon return from the debug code. If a SHDW bit is set to "1," a read from breakpoint register location will yield the contents of the breakpoint registers. If the SHDW bit is set to "0," a read from breakpoint register locations will yield the contents of device peripheral registers mapped 15 at these locations.

The ICKBUG register includes a PPDAT bit, which is the Program Space or Data Space Compare Select Bit. When the PPDAT bit is set to "0," the debug circuitry is in the "program break" mode, in which the value in the CNTBUG register is decremented every time the contents of the program memory address bus equals the contents of the 20 breakpoint address register (discussed later) on instruction fetch or a fetch of a first word of a two word instruction. A breakpoint will occur when CNTBUG underflows below

"0." When the PPDAT bit is "1," the debug circuitry is in "data break" mode, which means that the DATRW bit, see next paragraph, is enabled.

The DATRW bit is the Data Read/Write Access Select bit. If the PPDAT bit is one, the DATRW bit is set to "0," and the File Register Address (discussed below) equals 5 the breakpoint address on a read cycle, a break will result. If PPDAT is set to "1," DATRW is set to "1," and the File Register Address equals the breakpoint address on a write cycle, a break occurs. In either case, the SFR value read or written is copied into the GLOBUG register (discussed below). If the PPDAT bit equals "0," the DATRW bit is a don't care bit.

10           The DCMP bit is a data compare bit. If the PPDAT bit is set to "1," and the DCMP bit is set to "1," the contents of the data memory data bus are compared to the contents of the GLOBUG register. If the File Register Address equals the breakpoint address and the GLOBUG register equals the data being read or written, a breakpoint occurs. If PPDAT is set to "1" and DCMP is set to "0," no data comparison is done. If 15 PPDAT is set to "0," DCMP is a don't care bit.

The STKBRK bit is a Stack Overflow/Underflow Break bit. If the STKBRK bit is set to "1," a break occurs when the stack overflows or underflows, as represented by STKOVF or STKUNF bits, respectively. If the SJKBRK bit is set to "0," no action occurs on those conditions.

BKA19-BKA16 are Breakpoint Address bits. They are bits 19-16, respectively, of the breakpoint address register and bits 3-0 of the File Register.

The BIGBUG register includes BKA15-BKA8, which are bits 15-8, respectively, of the breakpoint address register and bits 7-0 of the File Register.

5       The GLOBUG register includes BKA07-BKA00 which are bits 7-0, respectively, of the breakpoint address register, or, if the PPDAT bit is set to "1," bits 7-0 of the breakpoint data register.

10      The CNTBUG register includes bits BKC07-BKC00, which are breakpoint counter bits. This register holds a counter value for the number of passes to allow before break is issued, as discussed below.

The SLGBUG register includes an RSBUG bit, which is the debugger reset vector selection. If the RSBUG bit equals one, all resets will vector to address 200028H. If RSBUG equals zero, all resets will vector to 000000H.

15      The debug logic circuit 400 and the registers 410, illustrated in more detail in Fig. 7, interface with the data memory data bus 700, the data memory address bus 702, and the program memory address bus 704, and produce the break signal 405, which is used to interrupt the CPU. The registers 410 include a breakpoint data register 410a, a breakpoint address register 410b, and a breakpoint count register 410c.

20      The debug logic circuit 400 includes a multiplexer 706, which has as inputs the data memory address bus 702 and the program memory address bus 704. The

multiplexer 706 produces one of these inputs on its output depending on a select signal 708, which is coupled to the PPDAT bit in the ICKBUG register. If the PPDAT bit is set to "1," the contents of the data memory address bus will appear at the output of the multiplexer 706. If the PPDAT bit is set to "0," the contents of the program memory address bus will appear at the output of the multiplexer 706.

The output of the multiplexer 706 is coupled to one input of an address comparator 710. The other input of the address comparator 710 is coupled to the breakpoint address register 410b. The address comparator 710 has an output which is high if the output of the multiplexer 706 equals the contents of the breakpoint address register 410b. Thus, the combination of the multiplexer 706 and the address comparator 410b can be used to compare either the contents of the data memory address bus 702 or the contents of the program memory address bus 704 to the contents of the breakpoint address register 410b.

The combination of the address comparator 710 and two 4-input AND gates 712, 714 forms a compare circuit 716 indicated by a dashed box in Fig. 7. The output of AND gate 712, an address-equal-on-read signal 718, is high when the READ signal from the CPU is high, the DATRW bit is low, the PPDAT bit is high, and the output of the address comparator 710 is high. Thus, the address-equal-on-read signal indicates that contents of the program memory address bus 704 (PPDAT is high, causing that bus to be selected by the multiplexer 706) matched the contents of the breakpoint address register 410b, on a read cycle when it was desired to look for such a match (DATRW high).

The output of AND gate 714, an address-equal-on-write signal 720, is high when the WRITE signal from the CPU is high, the DATRW bit is high, the PPDAT bit is high, and the output of the address comparator 710 is high. Thus, the address-equal-on-write signal indicates that contents of the program memory address bus 704 (PPDAT is high, 5 causing that bus to be selected by the multiplexer 706) matched the contents of the breakpoint address register 410b, on a write cycle when it was desired to look for such a match (DATRW high).

An OR gate 722 ORs the address-equal-on-read and address-equal-on-write signals. The resulting signal is ANDed with an inverted signal coupled to the DCMP bit 10 by AND gate 724. Thus, the output of AND gate 724 is high for either a read or a write when the contents of the program memory address bus 704 match the contents of the breakpoint address register 410, the PPDAT bit is high, and no data comparison is done (the DCMP bit is low). The output of the AND gate 724 is ORed by OR gate 726 to produce a breakpoint signal 728.

15 A second input to OR gate 726, and thus a second source of the breakpoint signal 728, is the output of AND gate 730. One of the inputs to the AND gate 730 is the output of a data comparator 732, which is high if the contents of the data memory data bus 700 equal the contents of the breakpoint data register 410a. Otherwise it is low.

A second input to the AND gate 730 is the DCMP signal. Thus, the output of the 20 AND gate 730 cannot be high unless the DCMP signal is high, indicating that a data compare is desired.

The third input to the AND gate 730 is the output of OR gate 722, which will be high under the circumstances described above.

Thus, the output of the OR gate 722 will be high, producing a breakpoint signal 728 at the output of OR gate 726, if (1) the DCMP bit is high (indicating that a data 5 compare is desired), (2) there is a match between the contents of the data memory data bus 700 and the breakpoint data register 410a, (3) there is a match on read (if DATRW is low) or write (if DATRW is high) between the contents of the program memory address bus 704 and the contents of the breakpoint address register 410b and (4) the PPDAT bit is high.

10 The third input to the OR gate 726, and thus a third source of the breakpoint signal 728, is the output of AND gate 734 which has as inputs the inverse of the signal coupled to the PPDAT bit and the output of the address comparator 710. This signal will be high when the contents of the data memory address bus equal the contents of the breakpoint address register 410b and PPDAT is low. In the exemplary embodiment 15 illustrated in Fig. 7, when PPDAT is low, the GLOBUG register is the breakpoint data register 410a and the breakpoint address register is only 12 bits.

The breakpoint signal 728 is used to downclock a breakpoint counter 730. In some debug applications (discussed below), the breakpoint counter 730 is loaded by storing a value in the breakpoint count register 410c and then asserting the 20 run-until-break-control signal 732. When the contents of the breakpoint counter 730 equal 00H and the down clock signal is asserted, an underflow condition will occur,

causing an underflow output of the breakpoint counter to be asserted, which produces the break signal 405.

In use, the personal computer 105 would store in the breakpoint count register 410c the number of passes by a particular breakpoint that are desired before interrupting 5 the CPU. The personal computer 105 would then cause the run-until-break-control signal 732 to be asserted, loading the breakpoint counter 730. Thereafter, every time that breakpoint is encountered, the breakpoint counter 730 would be decremented until it reaches zero. The next time the breakpoint is encountered, the breakpoint counter 730 would underflow and produce the break signal.

10 In one exemplary embodiment, the microcontroller is interrupted or halted when a breakpoint occurs. The typical HALT method is by breakpoint. To enter breakpoint mode, the BKBUG configuration bit is set to "0." When the PPDAT bit is "0," the ICKBUG, BIGBUG and GLOBUG registers contain a 20 bit value that is compared against PC (the value of the program counter). Bit zero of PC is not compared to the 15 breakpoint value because, in the exemplary embodiment, it is assumed to be zero. When the values are equal and INBUG bit is not set, the circuit will generate a break signal on that cycle.

If the CNTBUG register 00H, the system will break on the first occurrence of an address match.

The ICKBUG, BIGBUG and GLOBUG registers are mapped in an overlay of the user memory space. When the BKBUG bit is enabled, the breakpoint address registers can be read and written. The power off and master clear initialization state of these registers will be 000000h, equal to the reset vector.

5       Disabling the breakpoints is implemented by setting the breakpoint address to all ones (1FFFFFFH). When the breakpoint address is set to the last location of addressable program space, no breakpoint will occur as long as execution takes place in user program memory.

If the breakpoint is set to PC, the instruction at PC will be executed, and the stack  
10      will point to PC + 2 for the trap return. No manipulation of the stack is required for  
program address breakpoints.

The system provides the ability to break on program address match, as shown in Figures 8 and 9. Breakpoints of this type occur on an address match of the first address  
15      fetch of any two cycle instruction. A breakpoint on address match will typically not  
occur on an address match if the address is pointing to the second word of a two-word  
instruction or if the address is generated for TBLRD or TBLWT instructions, which treat  
program memory as data memory. For the case of conditional instructions, the second  
20      fetch of a two-cycle instruction points to a “dummy address,” so the breakpoint address is  
invalid. For two word instructions, the second fetch is only an operand. The breakpoint  
would typically not be set to an operand address in an application.

To initiate a breakpoint on program address match, the personal computer 105, through the protocol translator gateway 110 and the serial I/O 305, or the serial I/O shown in Figure 5, would first store the breakpoint address in the breakpoint address register 410b (block 800). This would cause data representing the desired breakpoint 5 address to be stored in bits BKA19-BKA00 in the ICKBUG register, the BIGBUG register and the GLOBUG register. The PPDAT bit in the ICKBUG register would then be set to zero (block 805), and the breakpoint count register would be set to zero (block 810). The target firmware would then be executed (block 815).

As the target firmware is being executed, the contents of the program memory 10 address bus 704 would be compared to the contents of the breakpoint address register 410b (block 900). If they are equal (block 905), and the breakpoint count register equals zero (block 910), which it is in this case, the CPU is interrupted (block 915). Thus, using this set of steps, the CPU can be interrupted when the PC equals a particular address.

The exemplary embodiment also supports breakpoints on a predetermined number 15 of program address matches. This type of breakpoint processing is very much like the processing discussed in reference to Figures 8 and 9. The difference is that the instruction is executed the predetermined number of times before the CPU is interrupted. To initiate this type of breakpoint processing, a 20 bit value to be compared to the PC is stored in the ICKBUG, BIGBUG and GLOBUG registers. The number of times that the 20 address this value represents is to be fetched (“passed”) is loaded into the CNTBUG register. Each time the breakpoint address register matches the PC, the value in

CNTBUG is decremented. Before the CNTBUG register is decremented, the value is compared to zero. If it is zero and the INBUG bit is not set, then the circuit will generate a halt signal on that cycle.

The value to be loaded into CNTBUG is equal to the number of passes to be  
5 allowed minus one. This insures backward compatibility with the standard breakpoint  
halt.

The CNTBUG register must be 00H to break on the first occurrence of an address  
match. Any other value in CNTBUG, with PPDAT bit equal to “0,” will invoke a  
passpoint operation and a halt will not occur until CNTBUG equals zero.

10 Processing to implement the break on number of program address matches is  
illustrated in Figures 10 and 11. The personal computer sets up this breakpoint mode by  
storing the breakpoint address in the breakpoint address register 410b (block 1000). The  
PPDAT bit is set to zero (block 1005), and the breakpoint count register 410b is set to  
equal the number of desired passes (block 1010). The target firmware is then executed  
15 (block 1015).

As the firmware executes, the contents of the program memory address bus 704  
are compared to the contents of the breakpoint address register 410b (block 1100 of  
Figure 11). A check is made to determine if the contents are equal (block 1105). If not,  
execution is looped back to block 1100. Otherwise (i.e., if the contents are equal) then  
20 the breakpoint count register is decremented (block 1110) and another check is made to

determine if the breakpoint count register is equal to zero (block 1115). If the breakpoint count register is not equal to zero, then execution jumps back to block 1100 and the cycle repeats. Otherwise, i.e., if the breakpoint count register is equal to zero, then the CPU is interrupted (block 1125).

5       The exemplary embodiment also provides the ability to halt execution on a Special Function Register address match. This breakpoint is selected by setting the PPDAT bit to “1.” The ICKBUG and BIGBUG registers are loaded with a 12 bit value that is compared against the output of the multiplexer 706, shown in Figure 7. Normally, the value in GLOBUG is ignored during an SFR access breakpoint. The type of access is  
10      configured by the DATARW bit. When the values are equal and INBUG is not set, the circuit will generate a halt signal on that cycle.

Alternatively, the DCMP bit may be set to cause a compare between the value of the written SFR data and the GLOBUG value. If PPDAT and DCMP are both “1” and the SFR address equals the contents of ICKBUG and BIGBUG and the SFR data equals  
15      GLOBUG, a halt signal will be issued in that cycle.

As before, the CNTBUG register will be set to zero if it is desired to break on the first occurrence of a SFR address or address/data match. Any other value in CNTBUG, with the PPDAT bit set to “1,” will invoke a pass point operation and a halt will not occur until the CNTBUG bit equals zero.

20       The DATARW and the TCMP bits have no effect if PPDAT equals zero.

Because of the delay in the execution of the instruction, the halt occurs on PC + 2 instead of PC. The instruction at PC + 2 will have been executed and the stack will be pointing to PC + 4 as the trap returns.

Break on SFR address match processing is illustrated in Figures 12 and 13. The  
5 personal computer sets up the break on SFR address match breakpoint by storing the  
breakpoint data address in the breakpoint address register 410b (block 1200). The  
PPDAT bit is set to one and the DCMP bit is set to zero (block 1205). The breakpoint  
count register is set to zero (block 1210). The target firmware is then executed (block  
1215).

10 As the target firmware is executing, the contents of the data memory address bus  
702 are compared to the contents of the breakpoint address register (block 1300). If they  
are equal (block 1305) and the breakpoint count register equals zero (block 1310), which  
it does in this case, the CPU is interrupted (block 1320).

To set up a breakpoint on SFR address and data match, as illustrated in Figures 14  
15 and 15, the personal computer stores the breakpoint data address in the breakpoint  
address register 410b (block 1400). The breakpoint data is then stored in the breakpoint  
data register 410a (block 1405). The PPDAT bit and the DCMP bit are then set to “1”  
(block 1410). The breakpoint count register is set to zero (block 1415) and the target  
firmware is executed (block 1420).

As the target firmware executes, the contents of the program memory address bus 704 is compared to contents of the breakpoint address register 410b (block 1500). If they are equal (block 1505), the contents data memory data bus 700 are compared to the contents of the breakpoint data register 410a (block 1510). If they are equal (block 5 1515), the CPU is interrupted (block 1520).

To set up a breakpoint on a number of SFR address and data matches, as illustrated in Figures 16 and 17, the personal computer stores the breakpoint data address in the breakpoint address register 410b (block 1600). The breakpoint data is then stored in the breakpoint data register (block 1605). The PPDAT and DCMP bits are then set to 10 “1” (block 1610). The breakpoint count register is set to the number of passes desired (block 1615), and the target firmware is executed (block 1620).

As the target firmware executes, the contents of the program memory address bus 704 are compared to the contents of the breakpoint address register 410b (block 1700). If they are equal (block 1705), the contents of the data memory data bus 700 are compared 15 to the contents of the breakpoint data register 410a (block 1710). If they are equal (block 1715), the breakpoint count register 410c is examined to determine if it is zero (block 1720). If it is not equal to zero (block 1725), the breakpoint count register 410c is decremented (block 1730) and the loop repeats. If the breakpoint count register 410c equals zero the CPU is interrupted (block 1735).

20 A breakpoint may also occur on stack overflow and underflow conditions. Under software control, the stack overflow and underflow can cause a device to vector to the

debug code. This is enabled by the STKBRK bit. If either the stack overflow or underflow bits are set, a force trap is generated on that cycle.

If both the STKBRK and the STVRE bit are set, an overflow or underflow will set the appropriate stack overflow or stack underflow bits and cause a trap to the debug  
5 handler.

When such a breakpoint occurs, the stack is already full. Therefore, the trap execution will overwrite the last return address in the stack with a trap return address. The stack will then contain the sequence that caused the stack overflow, so the user has visibility to the sequence. It is up to the debugger system code in the personal computer  
10 to correctly handle the subsequent execution. In most cases, the user will not continue with the code, but will debug and reset the device.

The stack overflow and stack underflow bits are not cleared until the user or debugger software clears them or a power on reset clears them.

The invention, therefore, is well adapted to carry out the objects and attain the ends  
15 and advantages mentioned, as well as others inherent therein. While the invention has been depicted, described, and is defined by reference to exemplary embodiments of the invention, such references do not imply a limitation on the invention, and no such limitation is to be inferred. The invention is capable of considerable modification,  
alternation, and equivalents in form and function, as will occur to those ordinarily skilled  
20 in the pertinent arts and having the benefit of this disclosure. The depicted and described

embodiments of the invention are exemplary only, and are not exhaustive of the scope of the invention. Consequently, the invention is intended to be limited only by the spirit and scope of the appended claims, giving full cognizance to equivalents in all respects.

10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
698  
699  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
798  
799  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
898  
899  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1788  
1789  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1798  
1799  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1879  
18